УДК 004.056.53

СИСТЕМА СБОРА, АНАЛИЗА И ВИЗУАЛИЗАЦИИ ДАННЫХ ОБ УСТРОЙСТВАХ В СЕТИ ИНТЕРНЕТ

Ш.Р. ДАВЛАТОВ, П.В. КУЧИНСКИЙ

Белорусский государственный университет информатики и радиоэлектроники, Республика Беларусь

Поступила в редакцию

Аннотация. Разработана система сбора, анализа и визуализации технической информации о подключенных устройствах к сети Интернет на базе платформы Censys и языка программирования Python. Данная система предназначена для первичного анализа данных из открытых источников, результаты которого могут быть использованы администраторами по информационной безопасности в автоматизированных системах для предотвращения потенциальных угроз со стороны злоумышленников. Разработка была апробирована на примере сведений о серверах, обслуживающих наиболее популярные Интернет-ресурсы Беларуси.

Ключевые слова: поисковая система Censys, анализ больших данных, уязвимость, информационная безопасность.

Abstract. This article describes the system for collecting, analyzing and visualizing of technical information about devices connected to the Internet based on the Censys platform. It was developed using Python programming language and it's intended for primary data analysis. The system can be used by security administrators to prevent potential threats in automated systems. This project was tested on the example of information about devices serving the most popular Internet resources in Belarus.

Keywords: Censys search engine, big data analysis, vulnerability, information security.

Doklady BGUIR. 2018, Vol. 116, No. 6, pp. 19-25
System for collecting, analyzing and visualizing data about devices in the Internet
Sh.R. Davlatov, P.V. Kuchynski

Введение

Известные всему миру поисковые системы, такие как Google, Yandex, Bing, Yahoo индексируют преимущественно веб-страницы, статьи, ссылки, изображения, аудио- и видеофайлы. Однако существует еще и другая категория сервисов (например, Shodan или Censys), которые выполняют поиск в «базе данных всех устройств в Интернете» [1, 2]. Основной принцип работы подобных систем заключается в отправлении ping-запросов на все публично доступные IP-адреса и протоколировании их откликов. В результате создается своеобразная карта Интернета, на которой можно искать любые устройства с сетевым интерфейсом или изучать характер распространения определенной уязвимости в программном обеспечении (ПО). Также можно выбирать несколько критериев поиска и фильтрации данных для мониторинга актуального состояния той или иной информационной системы. К примеру, в одном случае была обнаружена возможность взять под контроль систему управления аквапарком, а в другом – подключиться к оборудованию атомной электростанции. Добавим к этому сетевые терминалы, банкоматы, «умные дома» и все остальные устройства, подключенные к сети. Получается интереснейшая картина: каждый пользователь потенциально может удаленно подключиться к любым незащищенным устройствам в глобальной сети Интернета вещей. Обнаружение уязвимой IP-камеры в глобальной сети ставит под угрозу сохранность частной жизни небольшой группы людей, а вот незащищенная

медицинская аппаратура или, например, бортовой компьютер поезда способны навредить куда большому количеству людей, если злоумышленники получат доступ к данным системам [3]. Вот почему производители и системные администраторы подобных информационных систем (ИС) должны быть предельно внимательны к вопросам защиты информации и обеспечения безопасности подключенных устройств к сети IoT.

Целью данной работы была разработка системы сбора, анализа и визуализации технической информации о веб-серверах на основе данных из поисковой системы Censys. Для решения поставленной задачи была разработана комплексная система, предназначенная для первичного анализа данных на этапе проведения аудита безопасности ИС.

Механизм работы платформы Censys

Поисковая система Censys как открытый проект стала доступна в октябре 2015 года после презентации на 22-й конференции по безопасности компьютеров и коммуникаций ACM CCS [2]. Данная платформа работает 24 часа в сутки 7 дней в неделю, собирая техническую информацию о подключенных устройствах к глобальной сети Интернет. Отклики сетевых узлов на ping-запросы помогают идентифицировать ответившие устройства и собирать доступные сведения о них. Все данные обновляются ежедневно в ходе сканирования общедоступного адресного пространства IPv4 и первого миллиона доменов в рейтинге посещаемости сайтов. Censys также может выдать список устройств, не защищенных от какой-то известной угрозы из числа наиболее распространенных: например, от Heartbleed – ошибка в криптографическом ПО OpenSSL, позволяющей осуществлять несанкционированное чтение памяти на сервере или клиенте, в том числе для извлечения закрытого ключа шифрования. Censys предоставляет свои данные исследователям через общедоступную поисковую систему, интерфейс REST API, таблицы в формате Google Big Query и в виде скачиваемых баз данных [2].

Веб-интерфейс платформы позволяет выполнять полнотекстовый поиск и запрашивать любые структурированные поля и теги, созданные при сканировании и постобработке. Censys также поддерживает регулярные выражения и числовые диапазоны, а пользовательские запросы могут быть объединены с помощью логических структур (NOT, OR, AND). В качестве примера рассмотрим поисковой запрос, который выдает из базы платформы список всех доступных устройств, которые расположены на территории Беларуси:

location.country: Belarus AND 80.http.get.status_code: 200

Выражение location.country: Belarus возвращает список устройств на территории страны, а правая часть – 80.http.get.status_code: 200 означает, что найденные устройства должны иметь код состояния 200 (успешный HTTP запрос). В результате логической операции AND (И) получаем итоговый результат в виде пересечения двух списков, который удовлетворяет нашему критерию (см. рис. 1).

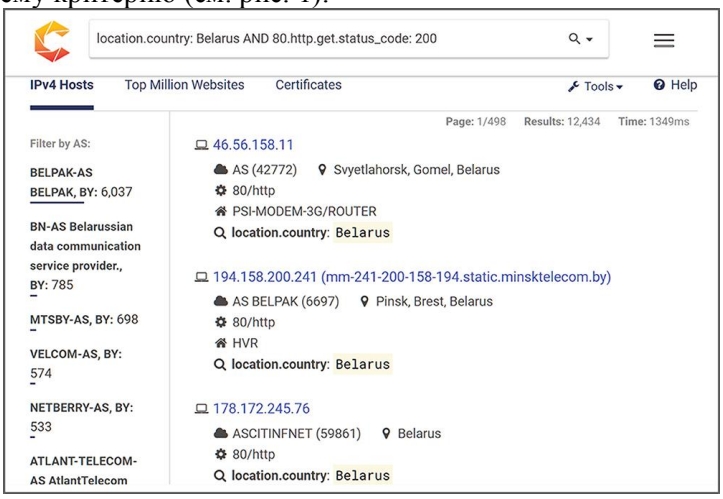

Рис. 1. Пример страницы результатов поиска в системе Censys

Архитектура разработанной системы и ее апробация

Система разработана на базе языка программирования высокого уровня Python, который имеет большое количество мощных библиотек для обработки и анализа данных. Архитектура всех модулей приложения построена на принципах объектно-ориентированного программирования (каждый модуль, как отдельный класс, инкапсулирует определенную логику). На рис. 2 приведена базовая схема работы системы, которая состоит из 4-х частей: модуля для сбора данных, локального хранилища, модулей для фильтрации, исследования и визуализации данных.

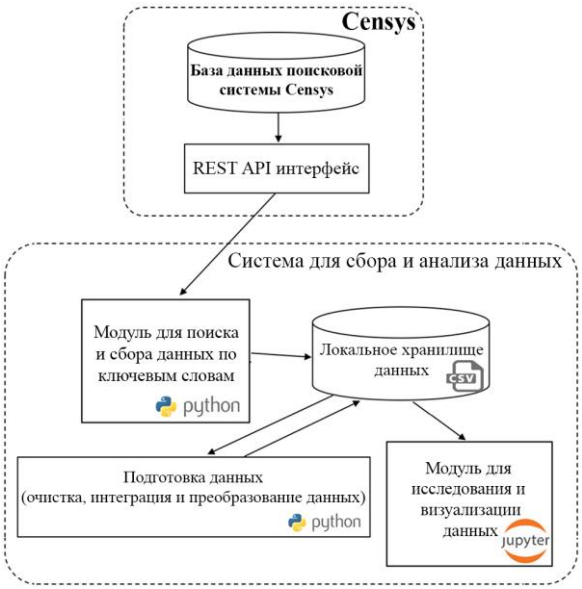

Рис. 2. Архитектура системы

В качестве апробации результатов разработанной системы была выбрана задача исследования данных о серверах, обслуживающих наиболее популярные Интернет-ресурсы Беларуси [4]. Для решения поставленной задачи был использован процесс Data Science [5], который состоит из следующих этапов: назначение цели исследования (данный этап уже был описан выше), сбор, подготовка, фильтрация, исследование и визуализация данных.

Сбор данных был осуществлен через REST API интерфейс системы Censys для получения технической информации об устройствах из базы данных. С помощью разработанного Python скрипта для каждого отдельного домена из нашего исходного списка сайтов осуществляется поиск всех доступных IP-адресов (см. рис. 3). На данном этапе мы уже получаем достаточно много полезной информации для дальнейшего процесса исследования: доменное имя, IP-адрес, страну физического расположения устройства, протоколы (в виде массива номеров и аббревиатуры портов), тип установленной операционной системы и серверного ПО. Промежуточные данные сохраняются в формате CSV (Comma-Separated Values, текстовый формат, предназначенный для представления табличных данных) для последующей обработки и исследования.

url	ip	location.country	protocols	metadata.os	80.http.get.headers.server
www.b2b.by/ru/	93.84.113.100	Belarus	[u'80/http', u'22/ssh', u'21/ftp']	Debian	nginx
www.b2b.by/ru/	86.57.234.188	Belarus	[u'80/http', u'25/smtp', u'993/imaps', u'143/i...	Windows	Microsoft-IIS/7.0
map.by	83.220.173.14	Russia	[u'80/http', u'110/pop3', u'995/pop3s', u'993/...	Debian	nginx
tut.by	83.220.173.14	Russia	[u'80/http', u'110/pop3', u'995/pop3s', u'993/...	Debian	nginx
21.by	94.130.0.108	Ukraine	[u'80/http', u'993/imaps', u'995/pop3s', u'25/...	Debian	nginx/1.2.1

Рис. 3. Первые 5 записей результатов поиска IP-адресов для каждого домена

Для каждого IP-адреса из таблицы (рис. 3) создается отдельный запрос в систему Censys для получения более детальной информации. В рамках политики ограничения использования данного сервиса, за 5 минут можно делать только 120 запросов на сервер для получения данных. Чтобы программа работала непрерывно, была реализована идея временной блокировки основного потока выполнения кода в течение 4–5 минут, после чего Python скрипт автоматически продолжает поиск информации по IP-адресам из списка. Преимущество данного подхода заключается в том, что непрерывно можно собирать все необходимые сведения об устройствах, которые соответствуют критерию поиска. Итого в нашей финальной выборке данных оказалось свыше 5000 записей, которые были сохранены в основной таблице локальной базы данных. Все остальные этапы обработки информации будут осуществляться на основе информации из данного хранилища (см. рис. 2).

Предварительный анализ данных из нашей выборки показал, что одному доменному имени в среднем соответствует 55 IP-адресов, на которых установлены различные службы (web, mail, ftp и другие серверные ПО). Заметим, что крупные Интернет-ресурсы с большой посещаемостью распределяют нагрузку на их системы за счет избыточности количества серверов с помощью управления ответами DNS-сервера в соответствии с определенной статистической моделью.

Следующий этап процесса Data Science – очистка и подготовка данных, который реализуется в отдельном модуле и обращается только к локальной базе. С помощью программной библиотеки Pandas для обработки данных проводится фильтрация всей собранной информации и ее хранение в структурированном виде для последующего процесса исследования [6]. Данный этап является очень важным, потому что ручная обработка данных (например, через табличный процессор Excel) требует больших временных затрат и существует вероятность допущения ошибки в процессе очистки данных. Основная идея заключается в том, чтобы подготовить и преобразовать «сырые данные» для последующей обработки с помощью Python скриптов и извлечь из них максимально полезную информацию. Следует отметить, что многие поля в ответе поисковых запросов являются пустыми, и таблицы данных без очистки оказались бы сильно разреженными. Зачастую это приводит к медленной работе программ, а также получению неточных оценок в моделях на этапе анализа и визуализации данных.

Рассмотрим пример колонки Protocols, где хранятся все открытые порты определенного устройства в виде массива строк (формат – номер/аббревиатура порта):

[‘80/http’, ‘22/ssh’, ’21/ftp’, ‘443/https’, ‘25/smtp’, ‘993/imaps’].

Система Censys возвращает данный массив в виде строковой переменной, следовательно, без предварительной обработки сложно анализировать данную характеристику устройства. Заметим, что на этапе визуализации мы также должны преобразовать подобный формат данных, потому что все функции построения графиков в качестве входных параметров принимают массив объектов. Следующий фрагмент кода позволяет преобразовать данную переменную в массив строк с помощью синтаксиса регулярных выражений [6, 7], где каждый элемент описывает определенный порт:

```
p = re.compile(“[0-9]+/[a-z]+”)
ports = p.findall(“[‘80/http’, ‘22/ssh’, ’21/ftp’, ‘443/https’, ‘25/smtp’, ‘993/imaps’]”).
```

После вышеперечисленных этапов мы имеем готовые данные для исследования и визуализации, которые хранятся в локальной базе данных в формате CSV (см. рис. 2). В качестве основного инструмента разработки используется Jupyter Notebook – удобная среда разработки для создания аналитических отчетов, так как позволяет хранить в одном файле исходный код программы, изображения, комментарии, формулы и графики. На этапе исследования было выявлено много интересных закономерностей в данных, результаты которых приведены ниже в виде наглядных иллюстраций.

Диаграмма на рис. 4 показывает распределение серверов из колонки “80.http.get.headers.server” основной таблицы базы данных. Она содержит информацию о типах веб-серверов как ПО, выполняющего функции приема/передачи HTTP запросов. Заметим, что лидирующие позиции занимают Apache и nginx, обслуживающие свыше 1300 серверов из нашей исходной выборки, что составляет примерно 27 %.

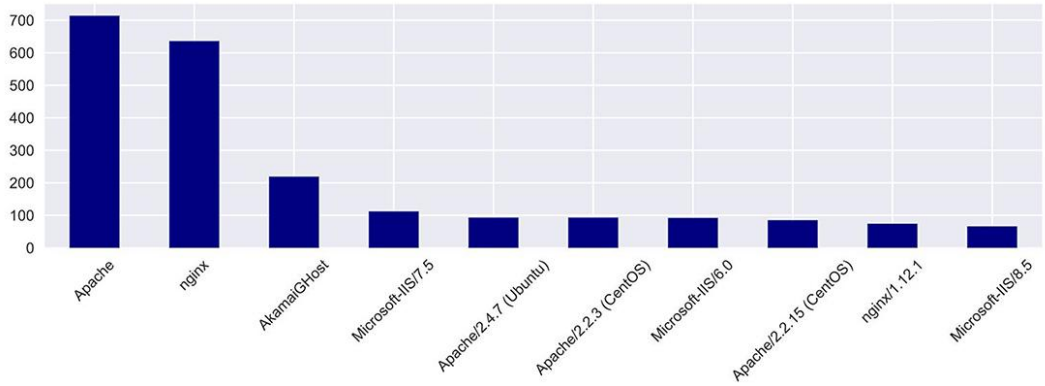

Рис. 4. Статистика распределения серверных ПО в выборке

На рис. 5 показана диаграмма распределения количества серверов по странам их физического расположения (первые 10 записей). Очевидно, что большинство крупных провайдеров услуг находятся на территории США, и поэтому большое количество серверов располагаются именно там. Важно подчеркнуть, что Беларусь занимает второе место в этом списке (примерно 14 % от общего объема данных в нашей выборке). Следовательно, большое количество владельцев Интернет-ресурсов предпочитают белорусские хостинговые компании для размещения своих веб-сайтов. Данный факт является не менее важным с точки зрения защиты данных и информационной безопасности в целом.

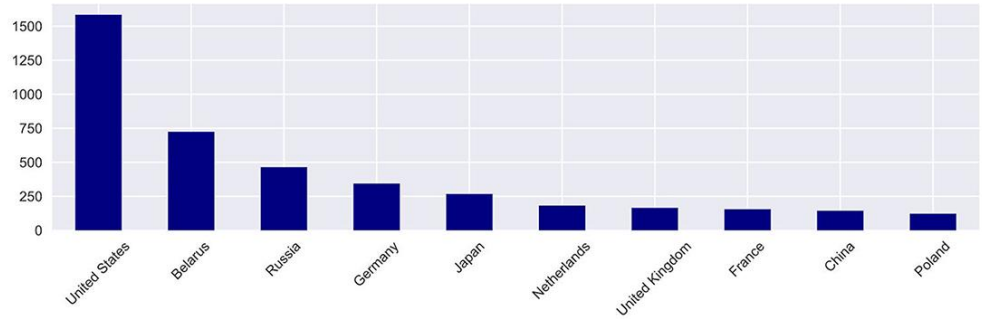

Рис. 5. Статистика распределения стран, по количеству физически расположенных серверов

На этапе анализа данных была создана тепловая карта, которая наглядно показывает корреляцию между данными об операционных системах (ОС) и странах расположения серверов (см. рис. 6).

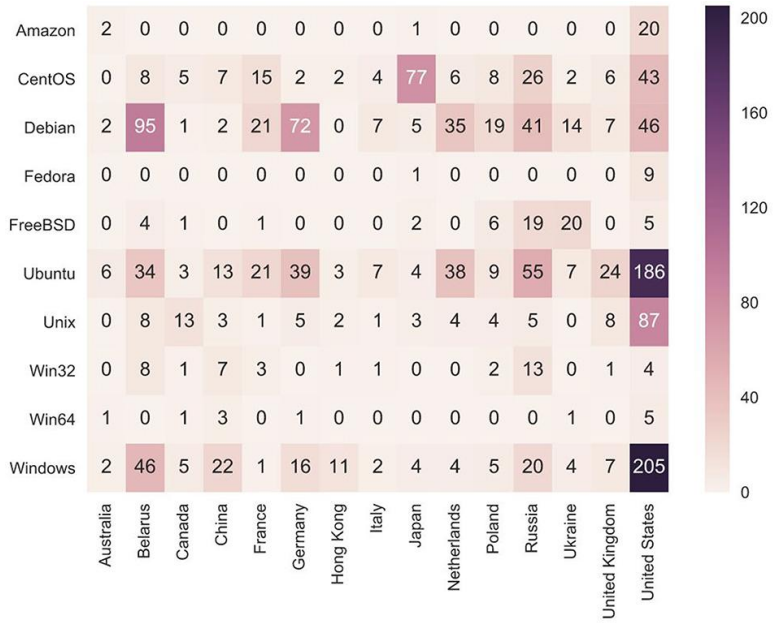

Рис. 6. Тепловая карта в формате [операционная система, страна расположения сервера]

Unix-подобные ОС (Debian, Ubuntu, CentOS и др.) гораздо чаще устанавливаются на серверах, где размещены Интернет-ресурсы из нашей исходной выборки. Заметим, что большинство оборудования, находящегося на территории США, работает под ОС Windows, в то время как на серверах, которые находятся в Беларуси, установлены в основном различные дистрибутивы ОС Linux. С точки зрения информационной безопасности это является важным фактором, поскольку системы данного семейства более надежные, распространяются в соответствии с моделью разработки свободного ПО и менее подвержены угрозам.

На основе данной диаграммы можно построить простейшие модели прогнозирования. Например, если мы знаем, что сервер находится на территории Беларуси, то с вероятностью 75 % на нем установлена ОС семейства Linux. А если предположить, что оборудование находится в США, то с вероятностью 60 % можно предположить, что на нем установлена ОС Windows. Следует отметить, что типичный процесс атаки на ИС начинается с процедур сбора и первичного анализа технической информации (серверное ПО, открытые порты, физическое расположение сервера и другие его характеристики). С учетом этих данных злоумышленники способны направить вектор атаки на более уязвимые места системы. Следовательно, на основе результатов исследования данной разработки специалисты по информационной безопасности могут проводить аудит безопасности и предотвратить потенциальные угрозы по конфиденциальности, целостности и доступности данных.

Заключение

Разработанная модульная система сбора, анализа и визуализации данных предназначена для проведения аудита безопасности ИС. Результаты исследования могут быть использованы специалистами по информационной безопасности в целях предотвращения потенциальных угроз со стороны злоумышленников на этапе проведения тестов на проникновение. В качестве основного источника информации была выбрана поисковая система Censys. Разработка апробирована на собранных данных о серверах, обслуживающих наиболее популярные Интернет-ресурсы Беларуси (вся информация находится в свободном доступе в базе данных поисковой системы www.censys.io). Следует отметить, что данная разработка обладает достаточной гибкостью, чтобы проводить детальный анализ данных и для других категорий устройств. Не менее важным преимуществом разработанной системы является возможность визуально отобразить результаты исследования в виде различных статистических графиков. Одним из основных достоинств системы является то, что имеется принципиальная возможность и эффективность применения персональных компьютеров малой и средней мощности для осуществления вышеперечисленного процесса сбора и анализа больших объемов данных. Это достигается путем применения оптимальных алгоритмов анализа данных с помощью высокопроизводительных библиотек языка Python.

Список литературы

1. Давлатов Ш. Сравнительный анализ инструментов для сбора и анализа данных из открытых источников OSINT // Матер. 53-й науч. конф. аспирантов, магистрантов и студентов БГУИР. Минск, 2–6 мая 2017 г.
2. Search Engine Backed by Internet-Wide Scanning / D. Zakir [et al.] // Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security. Denver, Colorado, USA, October 12–16, 2015. P. 572–553.
3. Скабцов Н. Аудит безопасности информационных систем. СПб.: Питер, 2018. 272 с.
4. Наиболее популярные белорусские интернет-ресурсы [Электронный ресурс]. URL: http://www.adsl.by/peering (дата обращения: 15.12.2017).
5. Davy C., Arno M., Mohamed A. Introducing Data Science: Big Data, Machine Learning, and more, using Python tools. Manning, 2016. 322 p.
6. Joel G. Data Science from Scratch. O'Reilly Media, 2015. 330 p.
7. Вандер П. Python для сложных задач. Наука о данных и машинное обучение. O'ReillyMedia, 2018. 576 с.

References

1. Davlatov Sh. Sravnitel'nyj analiz instrumentov dlja sbora i analiza dannyh iz otkrytyh istochnikov OSINT // Mater. 53-j nauch. konf. aspirantov, magistrantov i studentov BGUIR. Minsk, 2–6 maja 2017 g. (in Russ.)
2. Search Engine Backed by Internet-Wide Scanning / D. Zakir [et al.] // Proc. of the 22nd ACM SIGSAC Conference on Computer and Communications Security. Denver, Colorado, USA, October 12–16, 2015. P. 572–553.
3. Skabcov N. Audit bezopasnosti informacionnyh sistem. SPb.: Piter, 2018. 272 s. (in Russ.)
4. Naibolee populyarnye belorusskie internet-resursy [Electronic resource]. URL: http://www.adsl.by/peering (access date: 15.12.2017). (in Russ.)
5. Davy C., Arno M., Mohamed A. Introducing Data Science: Big Data, Machine Learning, and more, using Python tools. Manning, 2016. 322 p.
6. Joel G. Data Science from Scratch. O'Reilly Media, 2015. 330 p.
7. Vander P. Python dlja slozhnyh zadach. Nauka o dannyh i mashinnoe obuchenie. O'ReillyMedia, 2018. 576 s. (in Russ.)

Сведения об авторах

Давлатов Ш.Р., аспирант кафедры защиты информации Белорусского государственного университета информатики и радиоэлектроники.

Кучинский П.В., д.ф.-м.н., директор института прикладных физических проблем имени А.Н. Севченко.

Адрес для корреспонденции

220013, Республика Беларусь,
г. Минск, ул. П. Бровки, 6,
Белорусский государственный
университет информатики и радиоэлектроники
тел. +375-25-921-13-84;
e-mail: shohrukh.92@gmail.com
Давлатов Шохрух Рустамович

Information about the authors

Davlatov Sh.R., PG student of information security department of Belarusian state university of informatics and radioelectronics.

Kuczynski P.V., D.Sci, director of the Institute of applied physical problems named after A.N. Sevchenko.

Address for correspondence

220013, Republic of Belarus,
Minsk, P. Brovka st., 6,
Belarusian state university
of informatics and radioelectronics
tel. +375-25-921-13-84;
e-mail: shohrukh.92@gmail.com
Davlatov Shohrukh Rustamovich